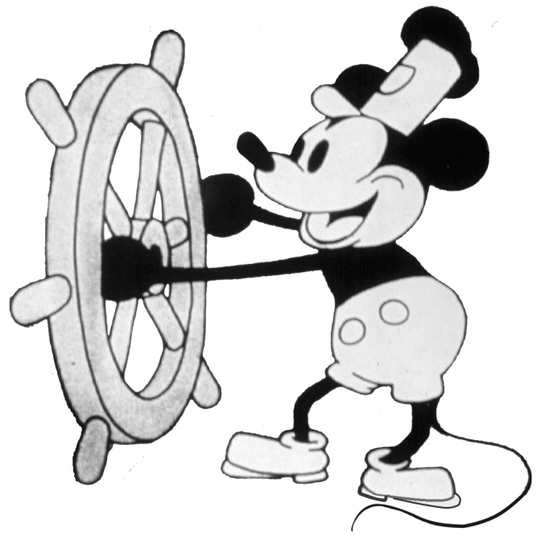


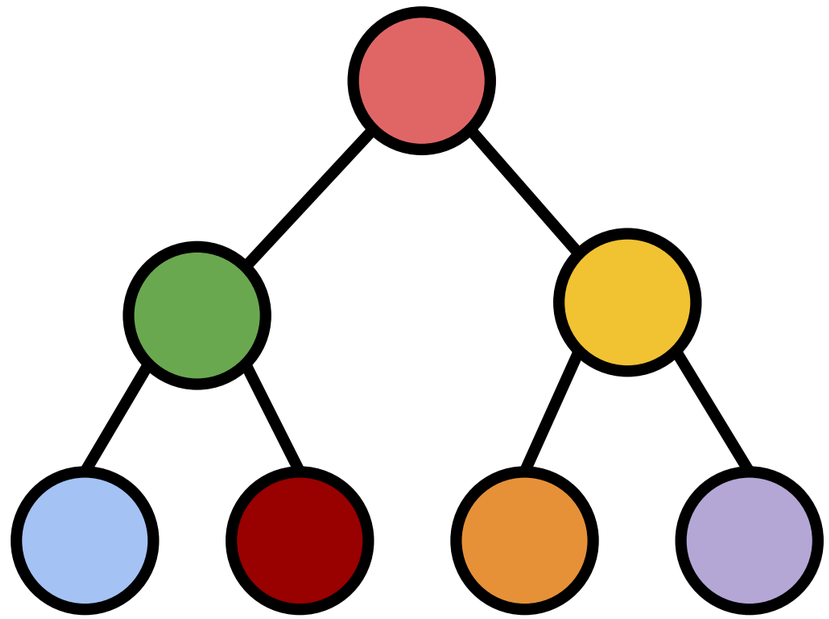
# CSCI 461: Computer Graphics

## Middlebury College, Fall 2025

---

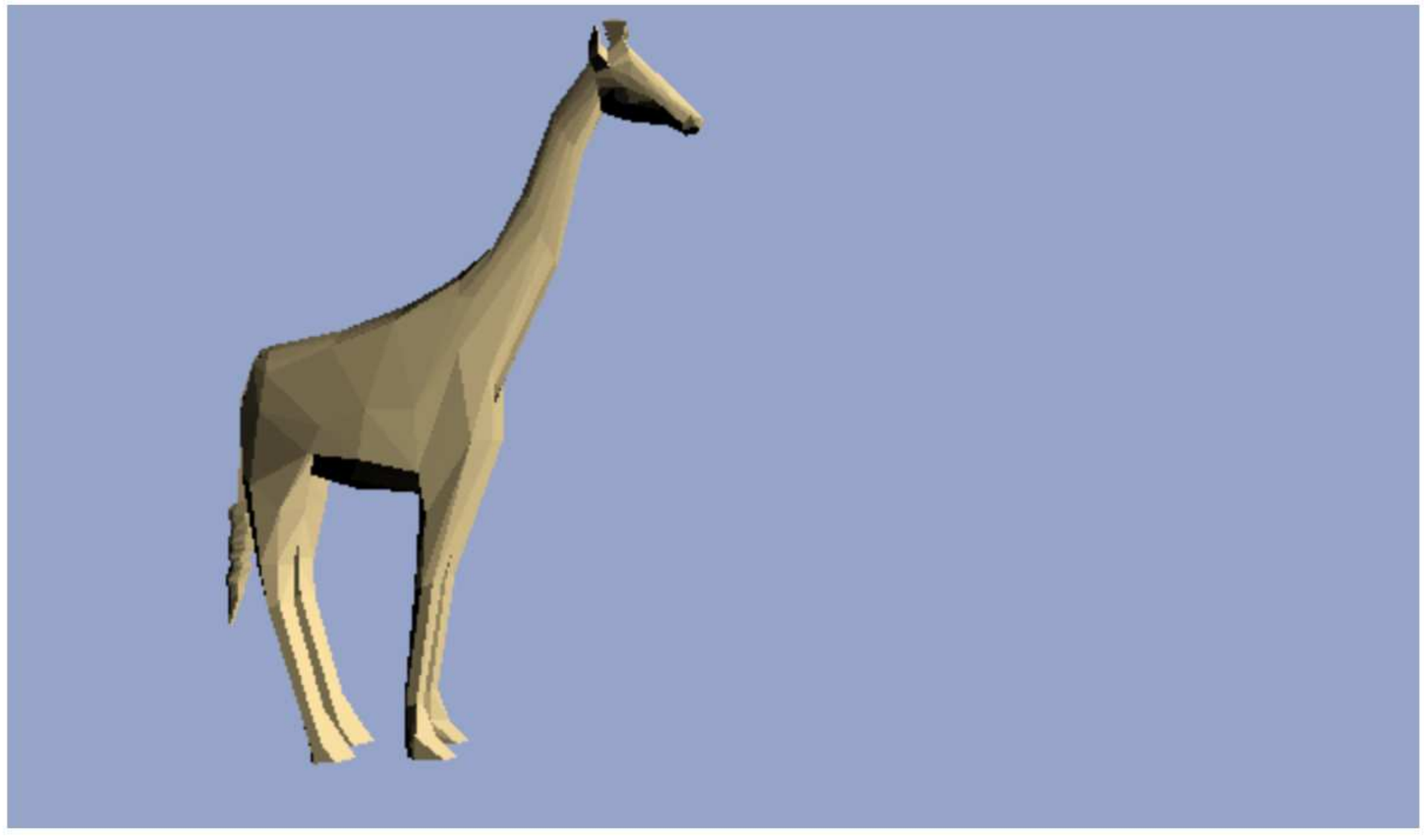
## Lecture 06: Acceleration Structures

**Use the Reactions page and pick the emoji which best describes how you feel about binary trees.**



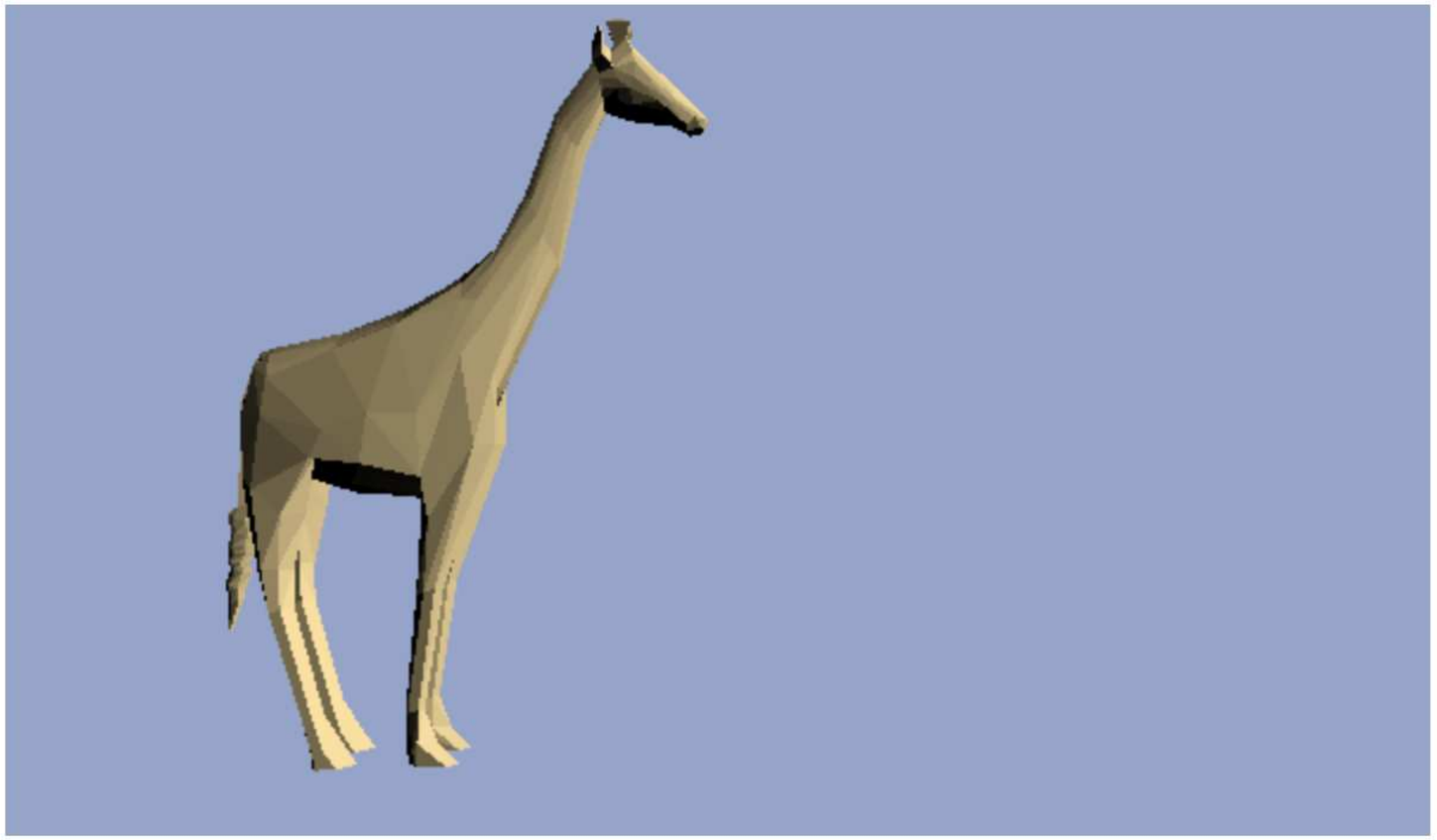
# By the end of today's lecture, you will be able to:

- intersect rays with **axis-aligned bounding boxes** (AABB),
- build a **Bounding Volume Hierarchy** (BVH) and use it to find ray-model intersections in logarithmic time,
- render a giraffe **quickly**!

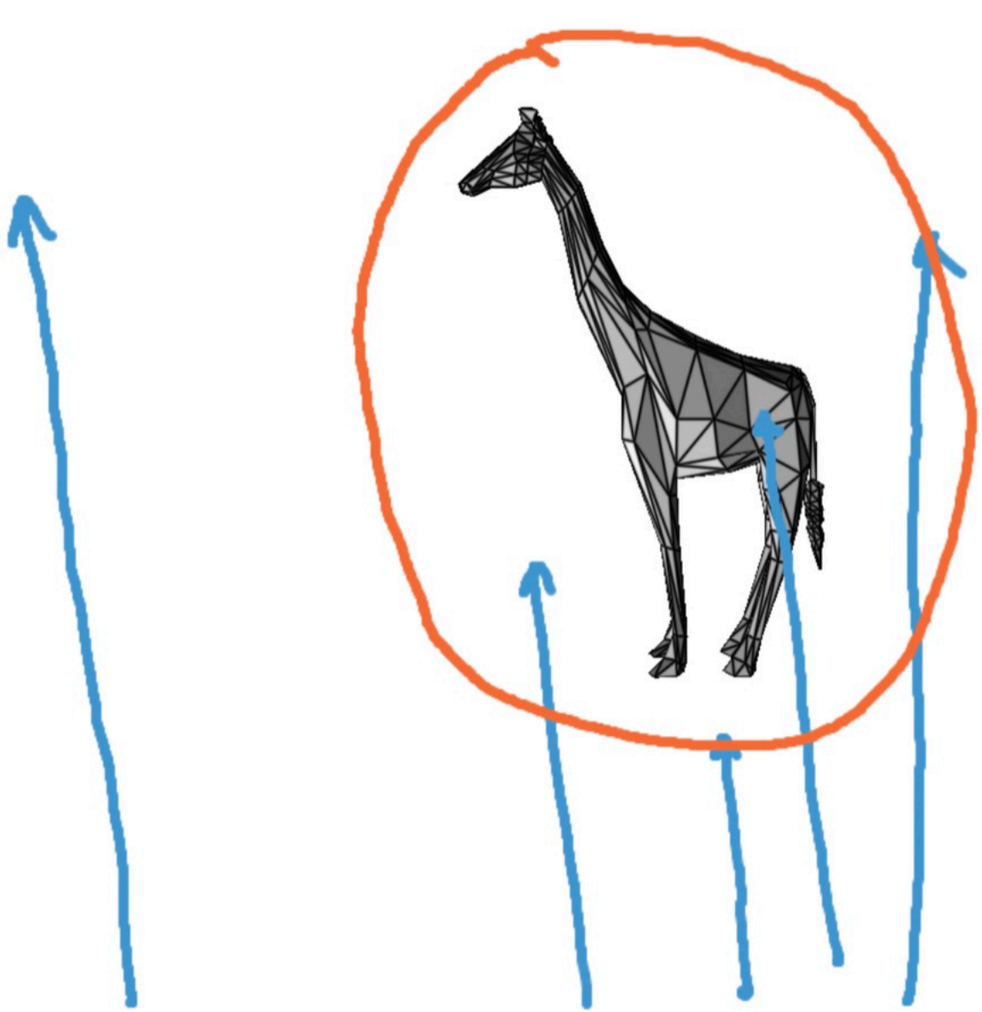


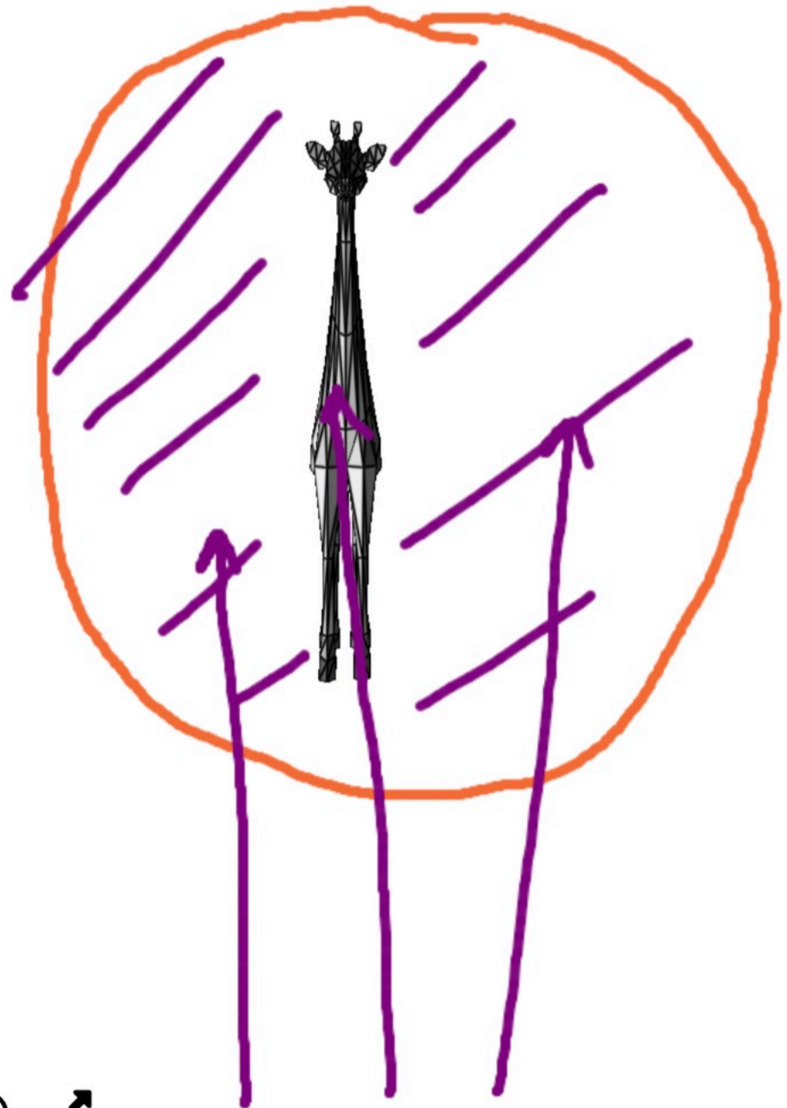
# The bottleneck of our ray tracers: ray-object intersections.

Click on `exercise` in row for today's class, accept assignment and `Go Live`.

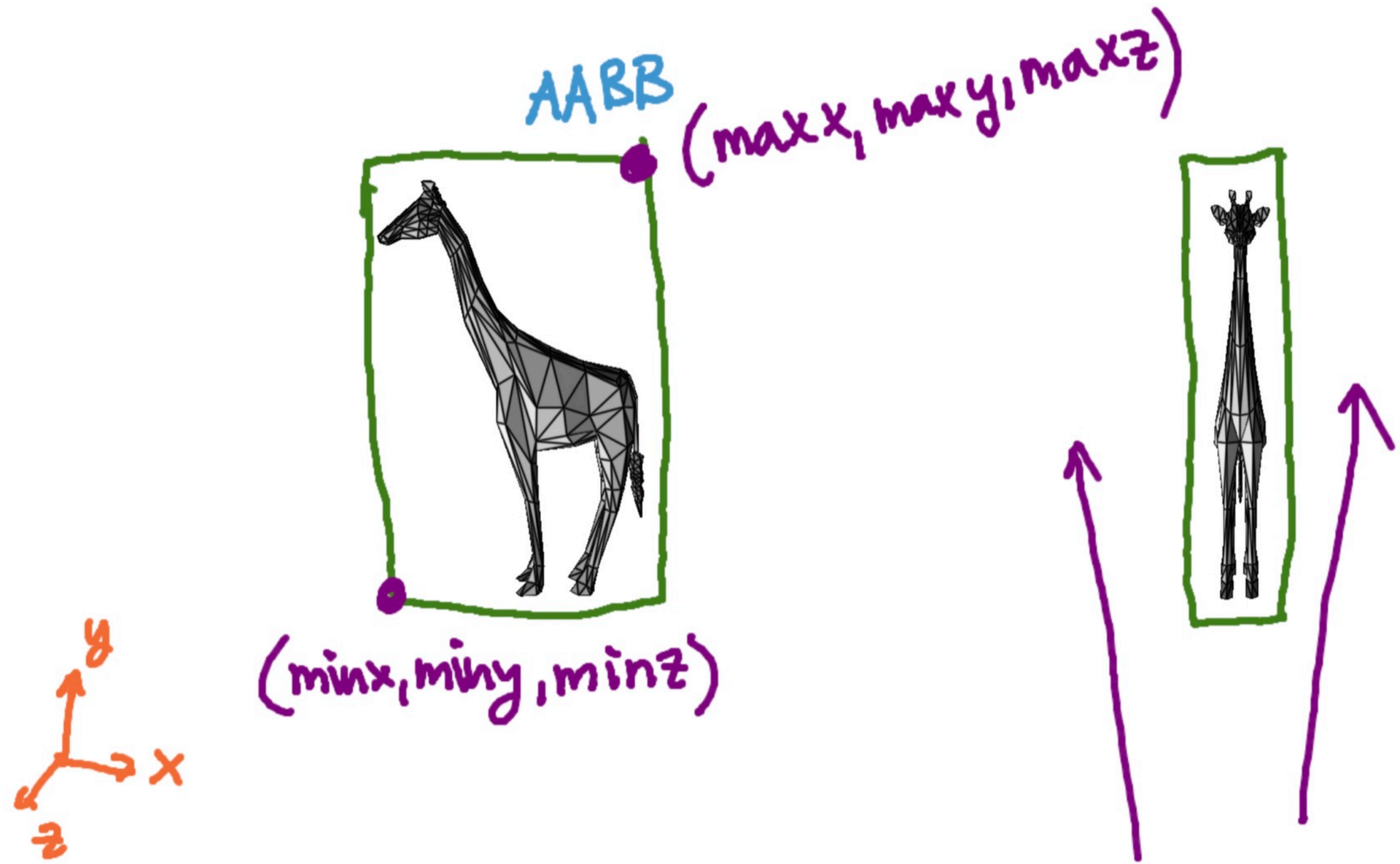


# Can we make it faster? Ideas?

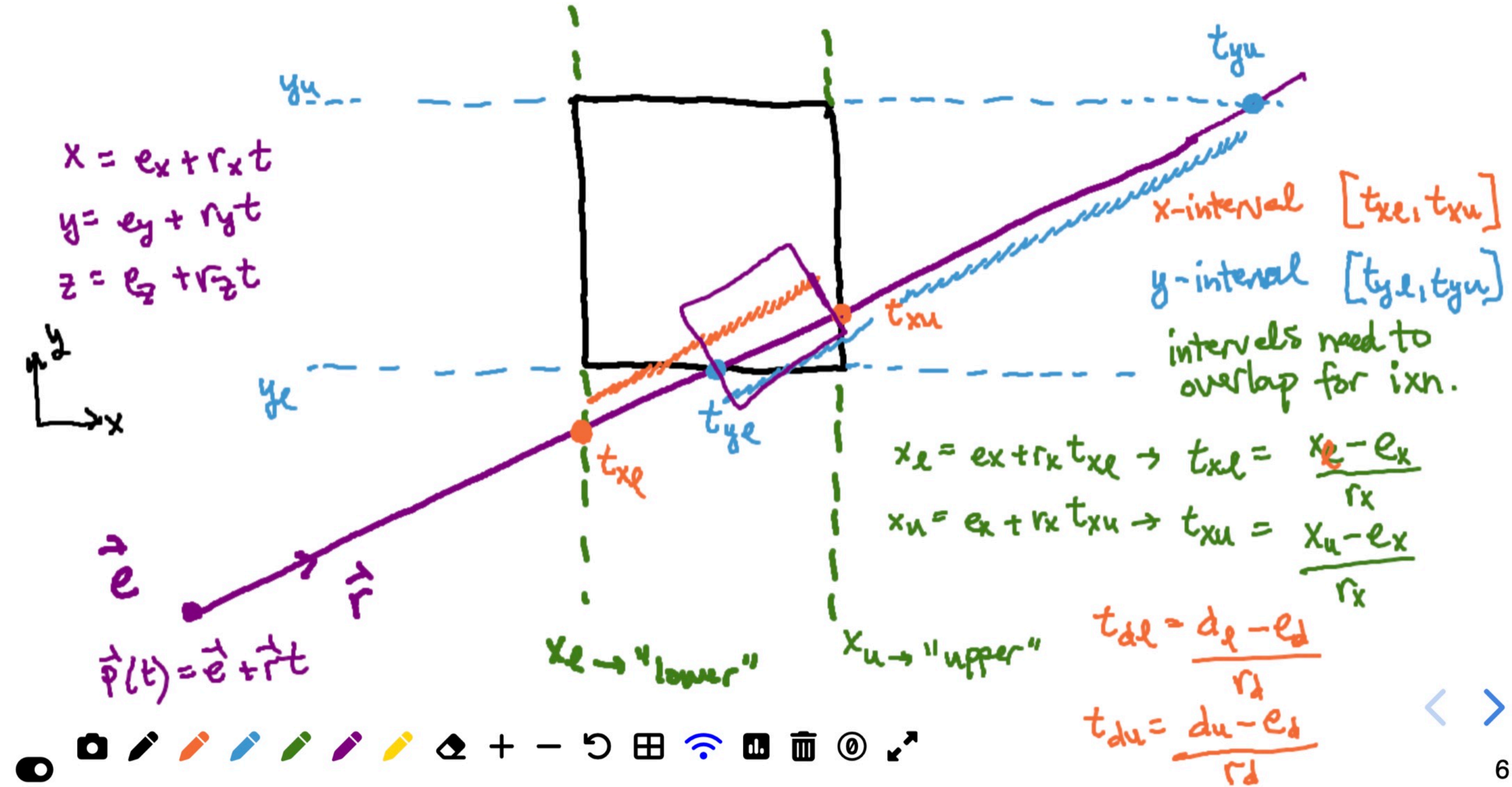




# Can we make it faster? Ideas?



# How to intersect a ray with an Axis-Aligned Bounding Box (AABB).



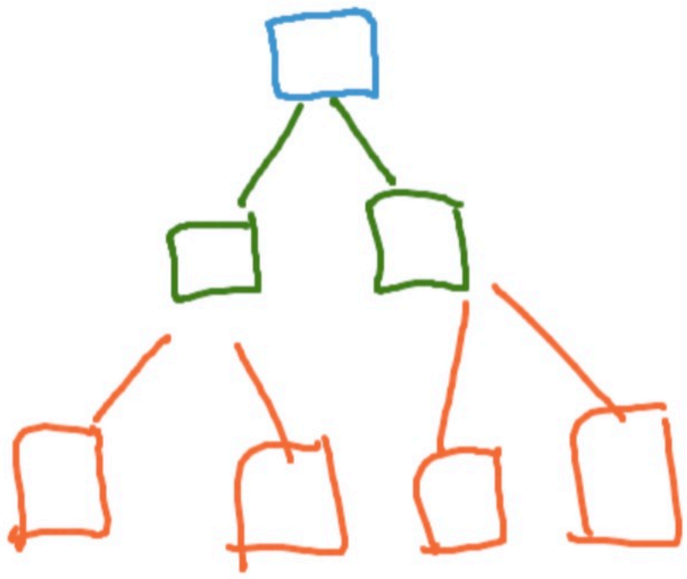
# Exercise 1: filter intersections using AABB of model.

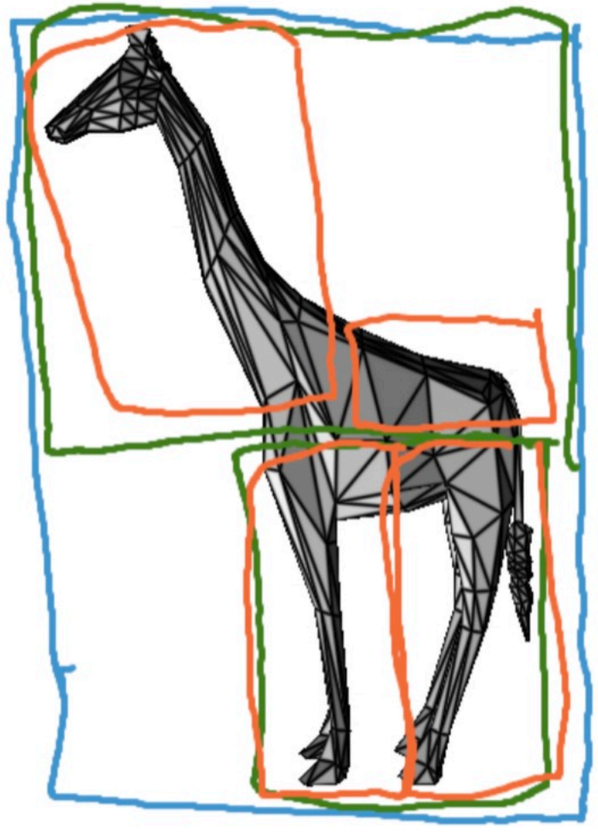
```
// in AABB class
intersect(ray, tmin, tmax) {
  // exercise 1a
  for (let d = 0; d < 3; ++d) {
    const tld = Math.min(
      (this.min[d] - ray.origin[d]) / ray.direction[d],
      (this.max[d] - ray.origin[d]) / ray.direction[d]
    );
    const tud = Math.max(
      (this.min[d] - ray.origin[d]) / ray.direction[d],
      (this.max[d] - ray.origin[d]) / ray.direction[d]
    );

    tmin = Math.max(tld, tmin);
    tmax = Math.min(tud, tmax);
    if (tmax <= tmin) return undefined;
  }
  return true;
}
```

```
// in "intersect" function of Model class
if (!this.box.intersect(ray, 1e-6, Infinity)) return undefined;
```

How much time now?

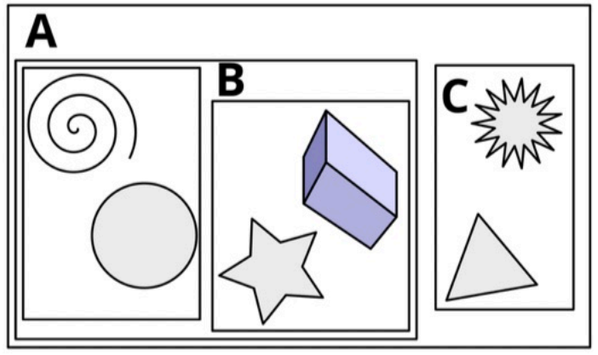
# Can we make this better?

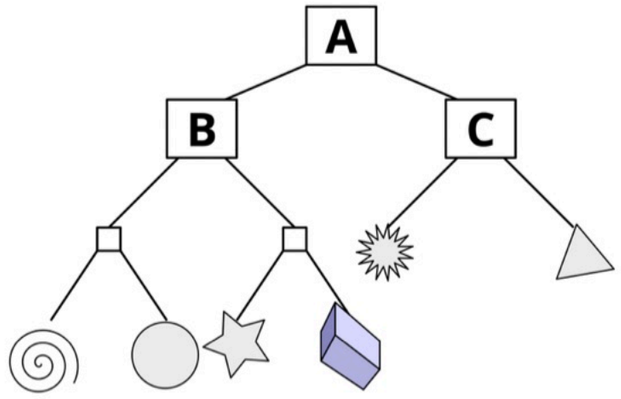




traverse
binary tree
of AABB
(each node is an AABB)

# How to build a Bounding Volume Hierarchy (BVH).




input: array of triangles (each triangle has an AABB)

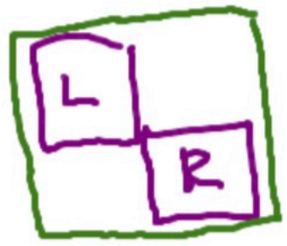
(from Wikipedia)

1.) sort the objects along some axis

2.) assign 1st half to left child node
assign 2nd half to right child node

3.) build the left and right subtrees

↳ enclose Left + rights AABB in this node's AABB



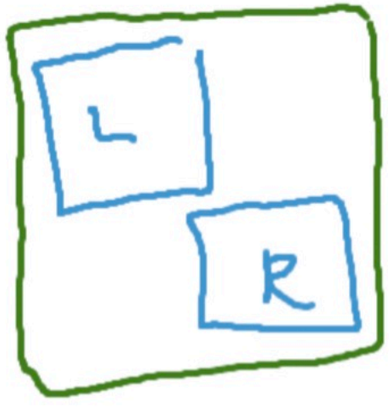
# Exercise 2A: complete BVH construction.

```
// BVHNode constructor
constructor(objects) {
  const axis = Math.floor(Math.random() * 3);
  const compare = (a, b) => {
    return a.box.min[axis] < b.box.min[axis];
  };
  let nObjects = objects.length;
  if (nObjects === 0) {
    return;
  } else if (nObjects === 1) {
    this.left = objects[0];
    this.right = objects[0];
  } else if (nObjects === 2) {
    if (compare(objects[0], objects[1])) {
      this.left = objects[0];
      this.right = objects[1];
    } else {
      this.left = objects[1];
      this.right = objects[0];
    }
  } else {
    objects.sort(compare);
    const mid = Math.floor(nObjects / 2);
    this.left = new BVHNode(objects.slice(0, mid));
    this.right = new BVHNode(objects.slice(mid, nObjects));
  }
  this.box = enclosingBox(this.left.box, this.right.box);
}
```

# Exercise 2B: calculating the `enclosingBox` of two `AABB`s.

```
const enclosingBox = (boxL, boxR) => {
  const lo = vec3.fromValues(
    Math.min(boxL.min[0], boxR.min[0]),
    Math.min(boxL.min[1], boxR.min[1]),
    Math.min(boxL.min[2], boxR.min[2])
  );

  const hi = vec3.fromValues(
    Math.max(boxL.max[0], boxR.max[0]),
    Math.max(boxL.max[1], boxR.max[1]),
    Math.max(boxL.max[2], boxR.max[2])
  );
  return new AABB(lo, hi);
};
```

# How to intersect a ray with a BVH node.




1.) check ixn with □

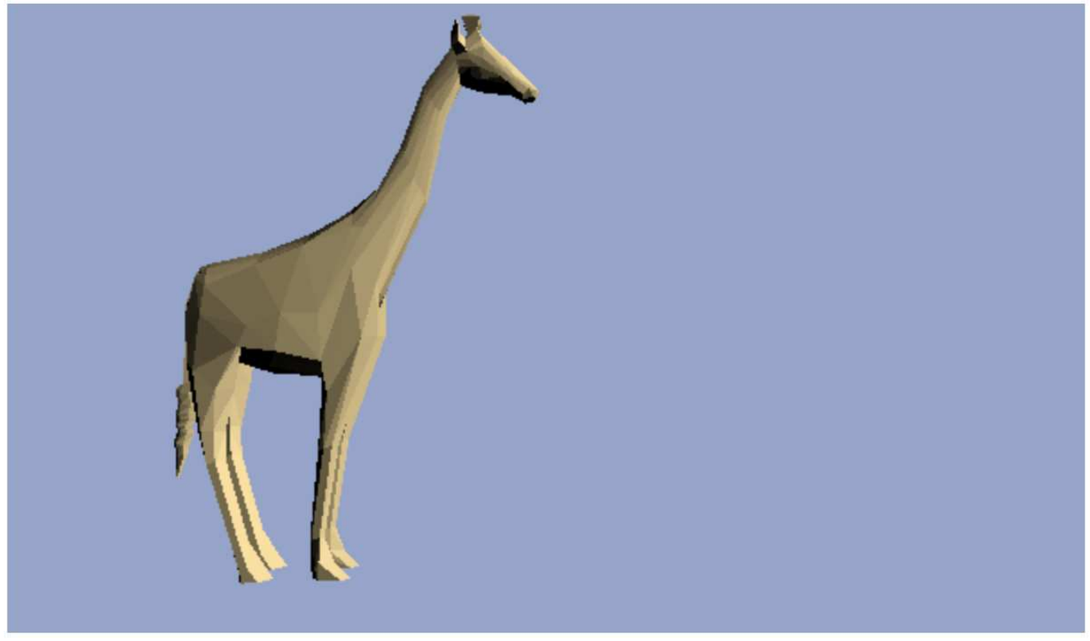
2.) check ixn with left box → tmin

3.) check ixn with right box

# Exercise 3: complete intersection of ray with BVH.

How much time now? What is the **speedup** over original value?

```
// in BVHNode "intersect" function
intersect(ray, tmin, tmax) {
  if (!this.box.intersect(ray, tmin, tmax)) return undefined;
  let ixnL = this.left.intersect(ray, tmin, tmax);
  let ixnR = this.right.intersect(ray, tmin, ixnL ? ixnL.t : tmax);
  return ixnR || ixnL;
}
```



# Summary & other options.

- **Kd-Tree:** partition space at each node instead of objects.
- **Parallelization:** every pixel can run independently of others.
- **Pre-lab 5** is a single question: find a `.obj` file you want to render!
- **Next week:** texturing.